EXAMENUL PENTRU OBȚINEREA ATESTATULUI PROFESIONAOLĂ

LA DISCIPLINA INFORMATICĂ

2017

SUBIECTE DE TIP II - PROGRAMARE

1. Fișierul text **numere.in** conține pe prima linie un număr natural **n (0<n<1000),** iar pe a doua linie **n** numere naturale cu cel mult 9 cifre fiecare, despărțite prin câte un spațiu. Scrieți un program C/C++ care citeşte toate numerele din fișierul **numere.in** şi care afişează în fişierul **numere.out** toate numerele din fișierul de intrare care au toate cifrele pare.

Exemplu:

numere.in	**numere.out**
4 23 24 555 2462	24 2462

```
#include <iostream>
#include <fstream>
using namespace std;
int par(int m)
{
    while(m%2==0 && m!=0)
        {
           m/=10;
        }
    if(m==0)
     return 1;
    else return 0;
}
int main()
{
    int n,i,x;
    ifstream f1("numere.in");
    ofstream f2("numere.out");
    f1>>n;

    for(i=1;i<=n;i++)
      {
          f1>>x;
         if(par(x)==1)
            {
               f2<<x<<' ';
            }

      }

    return 0;
}
```

2. Fișierul **atestat.in** conține pe prima linie un număr natural **n** de cel mult 9 cifre. Scrieți un program C++ care citește din „**atestat.in**” numărul **n** și care scrie în fișierul **atestat.out** pe prima linie, cifrele de rang par separate printr-un spațiu (rangul se numerotează de la dreapta la stânga începând cu cifra unităților care va avea rangul **1**); **Exemplu:**

Dacă fișierul **atestat.in** conține:
32245673 atunci **atestat.out** va avea următorul conținut:
7 5 2 3

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
   int n,i,x[1001],m=0;

   ifstream f1("atestat.in");
   ofstream f2("atestat.out");
   f1>>n;

   while(n)
   {
      x[++m]=n%10;
      n/=10;
   }
   for(i=2;i<=m;i+=2)
      f2<<x[i]<<' ';
   return 0;
}
```

3. În fișierul **date.in** pe prima linie este memorat un număr natural **n** iar pe următoarea linie sunt memorate **n** numere naturale separate prin spațiu, numere de maxim 9 cifre. Se cere să se afişeze pe prima linie a fișierului **date.out** cel mai mic număr impar și poziția primei apariții a acestuia. Dacă nu există astfel de numere să se afişeze mesajul "NU EXISTA"; **Exemplu:**

date.in	**date.out**
4 24 33 2343 33	33 2

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
   int n,i,x,poz=0,mini=100000;

   ifstream f1("date.in");
   ofstream f2("date.out");
   f1>>n;
   for(i=1;i<=n;i++)
    {
```

```
        f1>>x;
        if(x%2!=0)
          if(mini>x)
            {
             mini=x;
             poz=i;
             }
     }
     f2<<mini<<' '<<poz;

    return 0;
}
```

4. Fișierul **date.in** conține cel mult **50** de numere naturale, numerele fiind separate prin câte un spațiu. Fiecare număr are cel mult **4** cifre. Scrieți un program C/C++ care afișează pe câte o linie a fișierului **date.out**, fiecare număr citit, în ordinea citirii, şi divizorii proprii ai acestuia;
Exemplu: dacă fișierul **date.in** conține valorile:
23 12 100 36 atunci se va scrie în fișierul **date.out:**
23
12 2 3 4 6
100 2 4 5 10 20 25 50
36 2 3 4 6 9 12 18

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
    int n=1,i,x,d,nr=0,aux,dmax=0;
    ifstream f1("date.in");
    ofstream f2("date.out");

    while(f1>>x)
      {f2<<x<<' ';
        d=2;
          for(d=2;d<x;d++)
              if(x%d==0)
                f2<<d<<' ';
       f2<<endl;
      }

    return 0;
}
```

5. Fișierul text **Numere.in** conține cel mult 1000 de numere naturale cu cel mult patru cifre fiecare, despărțite prin câte un spațiu. Afișați pe ecran primul număr prim din fișier. Dacă fișierul nu conține nici un astfel de număr, atunci se va scrie mesajul NU EXISTA.
Exemplu: dacă fișierul **Numere.in** are următorul conținut:

300   11   44   25 7 13

pe ecran se va afişa **11**,

```
#include <iostream>
#include <fstream>
using namespace std;
int prim(int x)
{
   int d=2;
   while(d*d<=x && x%d!=0)
      d++;
   if(d*d>x)
      return 1;
   else return 0;
}
int main()
{
   int n=1,i,x,ok=0;
   ifstream f1("Numere.in");
   ofstream f2("Rezultate.out");
   f1>>n;
   for(i=1;i<=n;i++)
      {f1>>x;
         if(prim(x)==1)
            {cout<<x;i=n+1;ok=1;}
      }
   if(ok==0)
      cout<<"NU EXISTA";
   return 0;
}
```

6. În fişierul **numere.in** sunt memorate cel mult **1000** de numere naturale de cel mult 4 cifre fiecare.

Scrieţi un program care citeşte numerele din fişierul **numere.in** şi afişează pe prima linie a fişierului **numere.out** cel mai mare număr palindrom din fişierul **numere.in**; **Exemplu:**

numere.in	**numere.out**
10 101 23 44 676 43 88 201 202 124 141 90	676

```
#include <iostream>
#include <fstream>
using namespace std;
int palindrom(int x)
{int aux=x,inv=0;
   while(aux)
   {
      inv=aux%10+inv*10;
      aux/=10;
```

```
    }
    return x==inv;
 }
 int main()
 {
    int n=1,i,x,maxpal=0;
    ifstream f1("numere.in");
    ofstream f2("numere.out");
    f1>>n;
    for(i=1;i<=n;i++)
       {
          f1>>x;
          if(palindrom(x)==1 && x>maxpal)
             maxpal=x;
       }
       f2<<maxpal<<endl;

    return 0;
 }
```

7. Fişierul **atestat.in** conține două linii. Pe prima linie este scris un număr natural nenul **n, (5<n<30).** Pe cea de-a doua linie a fişierului sunt scrise **n** numere naturale separate prin câte un spațiu, formate fiecare din cel mult **4** cifre, reprezentând un şir de **n** numere naturale nenule. Cel puțin unul din cele **n** numere din şir este număr impar.
Să se realizeze un program în limbajul C/C++, care scrie în fişierul **atestat.out** pe prima linie, separate printr-un spațiu, numărul cifrelor numerelor impare din şir. **Exemplu:**

atestat.in	**atestat.out**
6 1232 243 98 3 242 27	3 1 2

```
#include <iostream>
#include <fstream>
int nrcifre(int m)
{int nr=0;
   while(m)
   {
      m/=10;
      nr++;
   }
   return nr;
}
using namespace std;
int main()
{
   int n=1,i,x;
   ifstream f1("atestat.in");
   ofstream f2("atestat.out");
   f1>>n;
   for(i=1;i<=n;i++)
```

```
    {
      f1>>x;
      if(x%2==1)
        f2<<nrcifre(x)<<' ';
    }


  return 0;
}
```

8. Din fişierul **tip1.in** se citeşte un număr natural **n** cu maxim nouă cifre nenule şi o cifră pară **k**. Să se scrie în fişierul **tip1.out** numărul care se obţine prin eliminarea tuturor cifrelor cu valoarea **k** din **x**; **Exemplu:**

Date de intrare: 1231 2

Date de ieşire: pe prima linie a fişierului **tip1.out** se va scrie 131

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
  int n,k,i=1,x[1000];
  ifstream f1("tip.in");
  ofstream f2("tip.out");

  f1>>n;f1>>k;
  int aux=n,m=1;

  while(aux!=0)
  {
   if((aux%10)!=k)
    {
     x[m]=aux%10;
     m++;
    }
    aux=aux/10;
  }

  for(i=m-1;i>=1;i--)
    f2<<x[i];
  return 0;
}
```

9. Fişierul **puncte.txt** conţine cel mult **1000** de linii, pe fiecare linie fiind scrise câte două numere întregi nenule semnificând coordonatele unor puncte în plan.
Scrieţi un program care citeşte coordonatele punctelor din fişierul **puncte.txt** şi afişează pe ecran numărul de valori care se află în cadranul **I**; **Exemplu:**

puncte.txt	**ecran**
1 1	2

10 1
-1 1

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
   int nr=0,x,y;

   ifstream f("puncte.txt");

   while(f>>x && f>>y)
   {
      if(x>=0 && y>=0)
          nr++;
   }
   cout<<"Numarul de puncte aflate in cadranul I : "<<nr;

   return 0;
}
```

10. Fișierul **fractii.txt** conține pe prima linie un număr **n** cu cel mult două cifre iar pe următoarele **n** linii câte două numere naturale nenule semnificând numărătorul și numitorul unei fracții. Scrieți un program care citește datele din fișierul **fractii.txt** și afișează pe ecran numărul de fracții ireductibile scrise în fișierul **fractii.txt**;

Exemplu:

Dacă fișierul **fractii.txt** are următorul conținut:

4
2 3
4 6
2 4 11
6

atunci pe ecran se va afișa valoarea: **2**

```
#include <iostream>
#include <fstream>
int cmmdc(int a,int b)
{int r;
   while(b)
   {
      r=a%b;
      a=b;
      b=r;
   }
   return a;
}
using namespace std;
int main()
{
   int n,a,b,nr=0,i;
```

```
   ifstream f("fractii.txt");
   f>>n;
   for(i=1;i<=n;i++)
   {
      f>>a;
      f>>b;
      if(cmmdc(a,b)==1)
         nr++;

   }

   cout<<nr<<' ';
   return 0;
}
```

11. Fişierul **atestat.in** conţine pe prima linie numărul natural nenul **n** (2<**n**<10). Pe fiecare dintre următoarele **n** linii sunt scrise câte trei numere reale **x , y** şi **r,** reprezentând coordonatele carteziene ale centrului unui cerc din planul xOy şi raza acestuia. Cele trei numere **x**, **y** şi **r** sunt separate prin câte un spaţiu.

Să se scrie un program, în limbajul /C/C++, care scrie în fişierul **atestat.out** coordonatele carteziene ale centrelor cercurilor din planul xOy şi raza acestora, sub forma (**x, y, r**) (abscisă, ordonată, rază), câte unul pe linie, în ordinea crescătoare a razelor; **Exemplu:**

atestat.in	**atestat.out**
5	(1,3,1)
1 3 1	(3,3,2)
4 4 7	(6,2,3)
2 5 5	(2,5,5)
3 3 2	(4,4,7)
6 2 3	

```
 #include <iostream>
#include <fstream>
using namespace std;
struct cerc
{
   int x,y,r;
}c[10];
int main()
{int i,n,j,aux;

   ifstream f("atestat.in");
   ofstream g("atestat.out");

   f>>n;
   for(i=1;i<=n;i++)
   {
      f>>c[i].x;
      f>>c[i].y;
```

```
      f>>c[i].r;
    }
    for(i=1;i<=n;i++)
      for(j=1;j<=n;j++)
        if(c[i].r<c[j].r)
          {
            aux=c[i].r;
            c[i].r=c[j].r;
            c[j].r=aux;

            aux=c[i].x;
            c[i].x=c[j].x;
            c[j].x=aux;

            aux=c[i].y;
            c[i].y=c[j].y;
            c[j].y=aux;
          }
        for(i=1;i<=n;i++)
          g<<"("<<c[i].x<<","<<c[i].y<<","<<c[i].r<<")"<<endl;
    return 0;
}
```

12. Fişierul text **numere.in** conţine pe prima linie un număr natural **n** (0<**n**<1000), iar pe a doua linie **n** numere naturale cu cel mult 9 cifre fiecare, despărţite prin câte un spaţiu. Scrieţi un program C/C++ care citeşte toate numerele din fişierul **numere.in** şi care, scrie pe prima linie în fişierul de ieşire **numere.out**, separate prin câte un spaţiu, numerele palindrom de pe a doua linie a fişierului **numere.in**; **Exemplu:**

numere.in	numere.out
8	53135 3113 373 239932 313
53135 3113 61556 373 54 239932 313 122	

```
 #include <iostream>
#include <fstream>
using namespace std;
int palindrom(int x)
{int aux=x,inv=0;
   while(aux)
   {
      inv=aux%10+inv*10;
      aux/=10;
   }
   return inv==x;
}
int main()
{int i,n,x;

   ifstream f("numere.in");
   ofstream g("numere.out");

   f>>n;
```

```
    for(i=1;i<=n;i++)
    {
       f>>x;
       if(palindrom(x)==1)
          g<<x<<' ';
    }


    return 0;
 }
```

13. Fişierul text **numere.in** conţine pe prima linie un număr natural **n (0<n<1000),** iar pe a doua linie **n** numere naturale cu cel mult 9 cifre fiecare, despărţite prin câte un spaţiu. Scrieţi un program C/C++ care citeşte toate numerele din fişierul **numere.in** şi care afişează pe prima linie în fişierul **numere.out** suma cifrelor fiecărui număr din fişierul de intrare; **Exemplu:**

numere.in	**numere.out**
3 82 500 1234	10 5 10

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{int i,n,s=0,x;

   ifstream f1("numere.in");
   ofstream f2("numere.out");

   f1>>n;
   for(i=1;i<=n;i++)
   {
      f1>>x;
      s=0;
      while(x)
      {
         s=x%10+s;
         x/=10;
      }
      f2<<s<<' ';
   }
   return 0;
}
```

14. Fişierul **atestat.in** conţine două linii. Pe prima linie este scris un număr natural **n** nenul (**n**<=50) iar pe a doua linie sunt scrise **n** numere naturale nenule de cel mult 9 cifre separate prin câte un spaţiu. Scrieţi un program C++ care citeşte din **atestat.in** numărul **n** şi apoi cele **n**

numere şi care scrie în fişierul **atestat.out** media aritmetică a numerelor de pe poziţii impare din şir;

Exemplu:

Dacă fişierul **atestat.in** conţine:

5

4 2 2 2 6 atunci **atestat.out** va avea următorul conţinut:

4

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{int i,n,x,ma=0,j=0;
   ifstream f("atestat.in");
   ofstream g("atestat.out");

   f>>n;
   for(i=1;i<=n;i++)
   {
      f>>x;
      if(i%2==1)
         {ma+=x;j++;}
   }
   g<<ma/j;
   return 0;
}
```

15. Fişierul text **numere.in** conţine pe prima linie un număr natural **n** (0<**n**<1000), iar pe a doua linie **n** numere naturale cu cel mult 9 cifre fiecare, despărţite prin câte un spaţiu. Scrieţi un program C/C++ care citeşte toate numerele din fişierul **numere.in**, construieşte un vector cu aceste numere, şi scrie pe prima linie în fişierul de ieşire **numere.out**, mesajul DA dacă numerele din vector sunt prime între ele două câte două şi mesajul NU în caz contrar; **Exemplu:**

numere.in	**numere.out**
7 5 131 56 14 55 456 513	NU

```
#include <iostream>
#include <fstream>
using namespace std;
int cmmdc(int a,int b)
{
   int r;
   while(b)
   {
      r=a%b;
      a=b;
      b=r;
   }
   return a;
```

```
}
int main()
{int x[1001],i,j,n,ok=1;

   ifstream f("numere.in");
   ofstream h("numere.out");
   f>>n;
   for(i=1;i<=n;i++)
      f>>x[i];
   for(i=1;i<=n;i++)
      for(j=i+1;j<=n;j++)
      {
         if(cmmdc(x[i],x[j])!=1)
            {h<<"NU";i=n+2;j=n+2;ok=0;}
      }

      if(ok==1)
         h<<"DA";

   return 0;
}
```

16. Numim **prefix** al unui număr natural **x** orice cât nenul obținut prin împărțirea întreagă a lui **x** la o putere a lui **10**.
Fişierul **numere.in** conține pe o singură linie un şir de cel mult **100** de numere naturale cu maxim **9** cifre, separate prin câte un singur spațiu.
Scrieți un program care citeşte valorile din fişierul **numere.in** şi scrie în fişierul **numere.out** o valoare ce reprezintă numărul de valori care au un prefix divizibil cu **10**; **Exemplu:**
Prefixele numărului **12546** sunt **12546, 1254, 125, 12, 1**. Dacă fişierul **numere.in** conține pe prima linie valorile:
103 234 567 613 61 100 345 2103 21 atunci fişierul
numere.out va avea următorul conținut:
3
deoarece sunt trei numere care au un prefix divizibil cu **10** şi anume **103**, **100** şi **2103**

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{int i,nr=0,x;

      ifstream f("numere.in");
      ofstream g("numere.out");
      while(f>>x)
      {
         while(x)
         {
            if(x%10==0)
               {nr++;x=0;}
            x/=10;
```

```
        }
      }
      g<<nr;
   return 0;
}
```

17. In fișierul **date.in**, pe prima linie, este memorat un număr natural **n**, iar pe linia următoare **n** numere naturale de cel puțin **3** cifre şi cel mult **9** cifre, cu toate cifrele nenule. Scrieți un program C/C++ care afişează pe prima linie a fişierului **date.out**, valorile obținute prin eliminarea primei cifre din fiecare dintre cele n numere citite;

Exemplu: dacă se citeşte pentru **n** valoarea **4** şi de pe rândul următor valorile

234 1567 278 5673

se obține în fişierul **date.out**:

34 567 78 673

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{int x,i,p=1,aux,n;

   ifstream f("date.in");
   ofstream g("date.out");

   f>>n;
   for(i=1;i<=n;i++)
   {
      p=1;
      f>>x;
      aux=x;
      while(aux)
      {
         aux/=10;
         p*=10;
      }

      g<<x%(p/10)<<' ';
   }
      return 0;
}
```

18. Fişierul **atestat.in** conține două linii. Pe prima linie este scris un număr natural nenul **n (5<n<30).** Pe cea de-a doua linie a fişierului sunt scrise **n** numere naturale separate prin câte un spațiu, formate fiecare din cel mult **4** cifre şi cel puțin **2** cifre, reprezentând un şir de **n** numere naturale. Cel puțin unul din cele **n** numere din şir este număr par.

Să se scrie un program, care preia valorile din fişierul **atestat.in** şi scrie în fişierul **atestat.out** separate printr-un spațiu, toate numerele **pare** din şir în **ordinea descrescătoare** a valorii lor;

Exemplu:

atestat.in
6
6138 12 496 2523 18 1243

atestat.out
6138 496 24 18 12

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{int i=0,n,j,x[31],a,aux,m=1;
   ifstream f1("atestat.in");
   ofstream f2("atestat.out");
   f1>>n;
   while(f1>>a)
   {
      if(a%2==0)
         x[m++]=a;

   }

   for(i=1;i<=m;i++)
      for(j=i;j<=m;j++)
         if(x[i]<x[j])
         {
               aux=x[i];
               x[i]=x[j];
               x[j]=aux;
         }

   for(i=1;i<m;i++)
      f2<<x[i]<<' ';

   return 0;
}
```

19. Fișierul text **numere.in** conține pe prima linie un număr natural **n (0<n<1000),** iar pe a doua linie **n** numere naturale cu cel mult 9 cifre fiecare, despărțite prin câte un spațiu. Scrieți un program care citește numerele din fișierul **numere.in**, verifică dacă numerele de pe a doua linie sunt în ordine crescătoare și scrie pe prima linie a fișierului **numere.out** unul din mesajele **Da** sau **Nu**.

Exemplu:

numere.in
12
10 301 23 44 676 43 88 201 702 124 147 90

numere.out
Nu

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{int i,n,x,y,ok=1;
```

```
   ifstream f1("numere.in");
   ofstream f2("numere.out");

   f1>>n;
   f1>>y;
   for(i=1;i<=n;i++)
   {
      x=y;
      f1>>y;
      if(x>y)
      {
         f2<<"Nu";
         i=n+1;
         ok=0;
      }
   }

   if(ok==1)
      f2<<"Da";


 return 0;
}
```

20. Fişierul **atestat.in** conține două linii. Pe prima linie este scris un număr natural nenul **n**, (5<**n**). Pe cea de-a doua linie a fişierului sunt scrise **n** numere naturale, separate prin câte un spațiu, formate fiecare din cel mult 9 cifre şi cel puțin 2 cifre, reprezentând un şir de **n** numere naturale. Primul număr din şir se găseşte pe poziția 1, iar ultimul număr din şir se găseşte pe poziția **n**. Cel puțin unul din cele **n** numere din şir are cifra zecilor impară. Să se scrie în fişierul **atestat.out**, separate printr-un spațiu, toate numerele din şir, care au cifra zecilor impară.

Exemplu:

Date de intrare :
6
6138 12 496 2523 18 1243
Date de ieşire:
6138 12 496 18

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{int i,n,x,aux;

   ifstream f1("atestat.in");
   ofstream f2("atestat.out");

   f1>>n;
```

```
   for(i=1;i<=n;i++)
   {
      f1>>x;
      aux=x;
      if((aux/10)%2)
         f2<<x<<' ';
   }

 return 0;
}
```

21. Se citește din fișierul **tip2.in** de pe prima linie un număr natural **n** iar pe a doua linie **n** numere naturale. Se cere să se afişeze în fişierul de ieşire **tip2.out**, valorile cu suma cifrelor un număr IMpar. **Exemplu :**

Date de intrare: 5
417 2152 26 664 7
Date de ieșire: 417 2152 664 7 ??? -_-???

```
#include <iostream>
#include <fstream>
using namespace std;
int cifracontrol(int a)
{
   if(a<=9)
      return a;
   if(a%9!=0)
      return a%9;
   else return 9;
}
int main()
{int i,n,x,s=0;

   ifstream f1("tip2.in");
   ofstream f2("tip2.out");
   f1>>n;

   for(i=1;i<=n;i++)
   {s=0;
      f1>>x;
      s=cifracontrol(x);
       if(s%2==1)
         f2<<x<<' ';

   }


 return 0;
}
```

22. Din fişierul text ***tablou.in*** se citesc: un număr natural **n** reprezentând numărul de componente ale unui vector cu numere naturale nenule cu cel mult 9 cifre fiecare şi cele **n** componente ale vectorului **(n<100).** Se cere să se afişeze elementele vectorului pe primul rând al fişierului text ***tablou.out***, ordonate crescător. **Exemplu:**

tablou.in	**tablou.out**
5 25 2 42 313 122	2 25 42 122 313

```
 #include <iostream>
#include <fstream>
using namespace std;
int main()
{int x[101],i,j,n,aux;

   ifstream f("tablou.in");
   ofstream h("tablou.out");
   f>>n;
   for(i=1;i<=n;i++)
      f>>x[i];

   for(i=1;i<=n;i++)
      for(j=i+1;j<=n;j++)
         if(x[i]>x[j])
         {
            aux=x[i];
            x[i]=x[j];
            x[j]=aux;
         }
   for(i=1;i<=n;i++)
      h<<x[i]<<' ';
   return 0;
}
```

23. Pe prima linie a fișierului **atestat.in** se află un număr natural n, iar pe următoarele n linii câte n numere naturale de maxim 2 cifre separate prin spații reprezentând elementele unui tablou bidimensional. Scrieţi un program C++ care citește din **atestat.in** numărul **n** și apoi elementele tabloului şi care scrie în fişierul **atestat.out** numărul de numere prime situate sub diagonala secundară; dacă nu este niciun număr prim se va scrie „**nu există**”. (un număr natural **n>=2** se numește **prim** dacă nu are alți divizori decât 1 și el însuși).

Exemplu :

Daca fişierul **atestat.in** conține:

4
6 4 4 1
2 6 8 1
0 4 2 2

8 6 7 4 atunci **atestat.out** va avea următorul conținut:
3

```
#include <iostream>
#include <fstream>
using namespace std;
int prim(int x)
{
   if(x==1)
      return 0;
   int d=2;
   while(d*d<=x && x%d!=0)
         d++;
   if(d*d>x)
      return 1;
   else return 0;
}
int main()
{int a[101][101],i,j,n,nr=0;

   ifstream f("atestat.in");
   ofstream g("atestat.out");
   f>>n;
   for(i=1;i<=n;i++)
      for(j=1;j<=n;j++)
        f>>a[i][j];

    for(i=2;i<=n;i++)
      for(j=n+2-i;j<=n;j++)
         if(prim(a[i][j])==1)
            {nr++;}

      if(nr==0)
         g<<"NU EXISTA";
      else g<<nr;

   return 0;
}
```

24. O matrice pătratică, **A**, de dimensiune **n**, cu **p** elemente naturale nenule, este memorată economic în fișierul de intrare **Matrice.txt** sub următoarea formă: pe prima linie a fișierului se găsesc două numere, **n și p**, dimensiunea matricei respectiv numărul de elemente nenule, iar pe următoarele **p** linii triplete de numere naturale (**v, l, c**) care reprezintă valoarea, linia respectiv

coloana pe care se găsesc elementele nenule. Scrieți un program care citește informațiile din fișierul de intrare, reface și scrie în fișierul de ieșire **Matrice.out** matricea **A**. **Exemplu:** dacă fișierul **Matrice.txt** are următorul conținut:

5 8
1 1 1
5 1 2
1 2 3
3 2 5
4 3 2
4 4 1
2 4 3 8 5 4 atunci fișierul **Matrice.out** va avea următorul conținut:

1 5 0 0 0
0 0 1 0 3
0 4 0 0 0
4 0 2 0 0
0 0 0 8 0

```
#include <iostream>
#include <fstream>
using namespace std;

int main()
{int a[100][100],i,j,n,p,v,l,c;
   ifstream f("Matrice.in");
   ofstream g("Matrice.out");
   f>>n;f>>p;
   for(i=1;i<=p;i++)
      {
         f>>v;
         f>>l;
         f>>c;
         a[l][c]=v;
      }

      for(i=1;i<=n;i++)
      {for(j=1;j<=n;j++)
         g<<a[i][j]<<' ';
         g<<endl;
      }
   f.close();
   g.close();

}
```

25. Fișierul **atestat.in** conține pe prima linie un număr natural **n (2<n<20)**, care reprezintă numărul de linii, respectiv numărul de coloane ale unei matrice. Pe fiecare din următoarele **n** linii, sunt scrise câte **n** numere întregi, formate din cel mult **4** cifre, separate prin câte un spațiu,

reprezentând valorile elementelor matricei. Să se scrie un program, care preia valorile din fişierul **atestat.in** şi scrie în fişierul **atestat.out**, separate printr-un spaţiu, **numărul** de elemente **prime** de pe fiecare **coloană** a matricei.

Liniile şi coloanele matricei vor fi numerotate începând cu **1**.

Exemplu:

atestat.in	**atestat.out**
3	**1 0 2**
7 1 4	1 1 1
4 6 2	
9 22 193	

//de cand e 2 neprim si ultimul rand cum de contine 2 prime ca eu vad doar pe 193//

```
#include <iostream>
#include <fstream>

using namespace std;
int prim(int x)
{
   int d=2;
   if(x==1)
      return 0;
   while(d*d<x && x%d!=0)
      d++;

   return d*d>x;
}
int main()

{int n,a,i,j,nr=0;
   ifstream f("atestat.in");
   ofstream g("atestat.out");
   f>>n;

   for(i=1;i<=n;i++)
   {for(j=1;j<=n;j++)
      {
         f>>a;
         if(prim(a))
            nr++;
      }
      g<<nr<<' ';
      nr=0;
   }
   g.close();
   return 0;
}
```

26. Fişierul text **matrice.in** conține pe prima linie un număr natural **n** **(0<n<20),** iar pe următoarele **n** linii câte **n** numere întregi de cel mult patru cifre, separate prin spațiu, reprezentând elementele unei matrice cu **n** linii şi **n** coloane. Scrieți un program care citeşte datele din fişierul **matrice.in** şi afişează în fişierul **matrice.out** cel mai mare număr situat pe diagonala secundară a matricei. **Exemplu:**

matrice.in	**matrice.out**
4 10 20 30 40 51 82 42 56 45 76 90 18 32 11 43 75	76

```
#include <iostream>
#include <fstream>
using namespace std;

int main()
{
 int n,mat,i,j,maxi;

 ifstream f("matrice.in");
 ofstream g("matrice.out");
   f>>n;
 for(i=1;i<=n;i++)
   for(j=1;j<=n;j++)
   {
      f>>mat;
      if(i+j==n+1 && mat>maxi)
         maxi=mat;
   }
   g<<maxi;
   f.close();
   g.close();
 return 0;
}
```

27. Fişierul **atestat.in** conține pe prima linie numărul natural nenul **n** (2<**n**<20), care reprezintă numărul de linii, respectiv numărul de coloane ale unei matrice **A**. Pe fiecare din următoarele **n** linii, sunt scrise câte **n** numere întregi, formate din cel mult 4 cifre, separate prin spații, reprezentând valorile elementelor matricei A. Să se scrie un program, în limbajul C/C++, care scrie în fişierul **atestat.out** numărul elementelor *numere perfecte* din matricea **A**; Un număr întreg a se numeşte *perfect* dacă este egal cu suma divizorilor mai mici decât el, de exemplu 6=1+2+3. Dacă nu există numere *perfecte* se va scrie mesajul "Nu exista numere perfecte". **Exemplu:**

atestat.in	atestat.out
3 4 9 4 2 35 57 -58 23 34 34 9 -5 69 6 26 33 28	~~1~~ 2

TROLOLOLOLO 6 SI 28 sunt numere PERFECTE

```
#include <iostream>
#include <fstream>
using namespace std;
int  perfect(int a)
{
   int d=1,S=0;
   while(d<a)
   {
      if(a%d==0)
         S+=d;
      return S==a;
   }
}
int main()
{
   int i,j,A,n,nr=0;

   ifstream f("atestat.in");
   ofstream g("atestat.out");
   f>>n;
   for(i=1;i<=n;i++)
      for(j=1;j<=n;j++)
      {
         f>>A;
         if(perfect(A)==1)
         {
            nr++;
         }
      }
      if(nr==0)
         g<<"Nu exista numere perfecte in fisier";
      else g<<nr;
   return 0;
}
```

28. Fișierul **atestat.in** conține pe prima linie un text de maxim 100 de caractere format din litere, cifre și spații. Cuvintele sunt separate prin câte un spațiu. Primul caracter este o literă.

Scrieți un program C++ care citește din **atestat.in** șirul de caractere și care scrie în fișierul **atestat.out** șirul din care s-au eliminat toate caracterele cifră; **Exemplu :**

Daca fișierul **atestat.in** conține:

Gigel are 2 mere si 3 alune

atunci fișierul **atestat.out** va avea următorul conținut:

Gigel are mere si alune

```
#include <iostream>
#include <cstring>
#include <fstream>
using namespace std;
int f[256];
int main()
{char s[205],voc[]="aeiou";
 int i=0,n=0,j=0,k=0;
 ifstream f1("atestat.in");
 ofstream f2("atestat.out");
f1.getline(s,205);
for(i=0;s[i];)
       {
          if(isdigit(s[i]))
            strcpy(s+i,s+i+1);
          else i++;

       }
f2<<s<<"\n";

 return 0;
}
```

29. Fișierul **atestat.in** conține o singură linie pe care se află scris un text format din cel mult **200** de caractere, primul caracter fiind literă, iar cuvintele din text sunt separate prin câte un spațiu. Fiecare cuvânt este format doar din litere mari sau mici ale alfabetului limbii engleze. Să se scrie un program care să afișeze textul pe o singură linie a ecranului, scris cu majuscule.

Exemplu:

atestat.in

Imi place Informatica

Ecran

IMI PLACE INFORMATICA

atestat.out

```
#include <iostream>
#include <cstring>
#include <fstream>
using namespace std;
int f[256];
int main()
{char s[205][100],voc[]="AEIOU";
```

```
 int i=0,n=0,j=0,nr=0;
 ifstream f1("atestat.in");
 ofstream f2("atestat.out");
 while(!f1.eof())
    {f1>>s[i];strupr(s[i]);cout<<s[i]<<' ';i++;}
     n=i;

for(i=0;i<n;i++)
    for(j=0;j<strlen(s[i]);j++)
       {
          if(!strchr(voc,s[i][j]))
                 nr++;
       }
f2<<nr;

 return 0;
}
```

30. Fișierul **atestat.in** conține o singură linie pe care se află scris un text format din cel mult **70** de caractere, primul caracter fiind literă, iar cuvintele din text sunt separate prin câte un spațiu. Fiecare cuvânt este format doar din litere mari sau mici ale alfabetului limbii engleze. Să se scrie un program, care preia valorile din fișierul **atestat.in** și afișează pe ecran **numărul** de apariții a primului caracter din text și apoi textul **transformat** astfel: litere mari în litere mici și litere mici în litere mari;

Exemplu:

atestat.in	**Date de ieșire**
<u>Ana</u> ateStaT colaC <u>elevele</u> feReaStra	**6 aNA ATEsTAt COLAc ELEVELE FErEAsTRA**

```
 #include <iostream>
#include <cstring>
#include <fstream>
using namespace std;
int main()
{char s[75][100],voc[]="aeiou",lit;
 int i=0,n=0,nr=0,j=0,ok=0,m;

 ifstream f1("atestat.in");
 ofstream f2("atestat.out");

 while(!f1.eof())
    {f1>>s[i];i++;}
    n=i;

  for(i=0;i<n;i++)
    for(j=0;j<strlen(s[i]);j++)
       {if(s[i][j]>='A' && s[i][j]<'a')
            s[i][j]+=32;
        else s[i][j]-=32;
       }
```

```
 lit=s[0][0];
  for(i=0;i<n;i++)
    for(j=0;j<strlen(s[i]);j++)
        if(s[i][j]==s[0][0] || s[i][j]==s[0][0]-32)
                nr++;

        cout<<nr-1<<' ';
    for(i=0;i<n;i++)
        cout<<s[i]<<' ';


 return 0;
}
```